International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 6, Issue 1, Feb 2016, 19-28

TRANS STELLAR
•Journal Publications •Research Consultancy

COMPARATIVE STUDY OF DIMENSIONALITY REDUCTION TECHNIQUES IN THE CONTEXT OF CLUSTERING

T. SUDHA[1] & P. NAGENDRAKUMAR[2]

[1]*Professor, Department of Computer Science, Sri Padmavathi Mahila University, Tirupati, Andhra Pradesh, India*

[2]*Research Scholar, Department of Computer Science, Vikrama Simhapuri University, SPSR Nellore, Andhra Pradesh, India*

ABSTRACT

Data mining has become one of the most recent prominent areas of research. Clustering is one of the main functionalities of data mining. Clustering high dimensional data often suffers from curse of dimensionality. The objective of this study is to make a comparison of cluster analysis respective to three dimensionality reduction techniques such as singular value decomposition, principal component analysis and multidimensional scaling. Cluster analysis has been performed on the original data as well as on all of the three lower dimensional data obtained through singular value decomposition, principal component analysis and multidimensional scaling using K-Means algorithm with varying number of clusters. Mean squared error and time have been considered as parameters for comparison. The results obtained show that the mean squared error obtained from the original data is almost same as the mean squared error obtained on the data reduced through multidimensional scaling but the order of the values differ due to random selection of cluster centers. It is also observed that the time taken for cluster analysis on the data reduced through singular value decomposition is less than the time taken for cluster analysis on the data reduced through principal component analysis and multidimensional scaling.

KEYWORDS: Clustering, Singular Value Decomposition, Principal Component Analysis, Multidimensional Scaling

Received: Dec 14, 2015; **Accepted:** Dec 18, 2015; **Published:** Dec 29, 2015; **Paper Id.:** IJCSEITRFEB20163

Case Study

INTRODUCTION

Data mining refers to the process of discovering interesting and useful patterns in large volumes of data. The different functionalities of data mining are class/concept description, association analysis, classification, clustering, outlier analysis and evolution analysis. Clustering is a common technique for statistical data analysis .Clustering is the process of grouping objects based on the principle of maximizing intra cluster similarity and minimizing inter cluster similarity. It is used in many fields, including machine learning, pattern recognition, image analysis, information retrieval and bioinformatics. Cluster analysis can be performed well when the number of dimensions in the given data is less.

The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high dimensional spaces that do not occur in low-dimensional spaces. When the number of dimensions in the given data increases, the volume of the space also increases. Dimensionality reduction can be considered as a

solution to deal with curse of dimensionality. Dimensionality reduction refers to the process of reducing the number of attributes under consideration.

In mathematical terms, dimensionality reduction can be stated as: Given the p-dimensional random variable $x=(x_1,x_2,.....x_p)^T$, find a lower dimensional representation of it, $s=(s_1,s_2,....s_k)^T$ with k<=p, that captures the content in the original data according to some criterion. Many dimensionality reduction techniques such as singular value decomposition, principal component analysis, multidimensional scaling, factor analysis, projection pursuit, independent component analysis etc have been developed. Applications of Dimensionality reduction include customer relationship management, text mining, image retrieval, micro array data analysis, protein classification; face recognition, handwritten digit recognition and intrusion detection. In this study, dimensionality reduction techniques such as singular value decomposition, principal component analysis and multidimensional scaling have been considered to reduce the high dimensional data in to low dimensional data.

Singular Value Decomposition (SVD)

It is a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. SVD is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices.

- An orthogonal matrix U
- A diagonal matrix S
- The transpose of the orthogonal matrix V

The theorem is usually represented as

$A_{mn} = U_{mn} S_{mn} V^T_{nn}$

Where $UU^T = I$ $V^TV = I$

The columns of U are orthonormal Eigen vectors of AA^T

The columns of V are orthonormal Eigen vectors of A^TA

S is a diagonal matrix containing the square roots of Eigen values from U or V in descending order.

Principal Component Analysis

It is a way of identifying patterns in data and expressing the data in such a way as to highlight their similarities and differences.

The steps of Principal Component Analysis are:

- Get some data.
- Subtract the mean.
- Calculate covariance matrix.
- Calculate the Eigen vectors and Eigen values of the covariance matrix.

- Choose components and form a feature vector.

  Feature vector = (eigen$_1$,eigen$_2$,…..eigen$_n$)

- Derive the new data set.

Multidimensional Scaling

Multidimensional scaling (MDS) is a set of related statistical techniques often used in information visualization for exploring similarities or dissimilarities in data. MDS is a special case of ordination. An MDS algorithm starts with a matrix of item–item similarities, and then assigns a location to each item in *N*-dimensional space, where *N* is specified a priori. For sufficiently small *N*, the resulting locations may be displayed in a graph or 3D visualization.

The data to be analyzed is a collection of *I* objects (colors, faces, stocks, . . .) on which a distance function is defined,

$\delta_{i,j}$:= distance between ith and jth objects.

These distances are the entries of the dissimilarity matrix

$$\Delta := \begin{pmatrix} \delta_{1,1}\delta_{1,2}\cdots\delta_{1,I} \\ \delta_{2,1}\delta_{2,2}\cdots\delta_{2,I} \\ \vdots \\ \vdots \\ \delta_{I,1}\delta_{I,2}\cdots\delta_{I,I} \end{pmatrix}$$

The goal of MDS is, given Δ, to find ***I*** vectors $x_1,\ldots\ldots\ldots x_I \in R^N$ such that

$$\left\| x_i - x_j \right\| \approx \delta_{i,j} \text{ for all } i, j \in I ,$$

Where $\|.\|$ is a vector norm. In classical MDS, this norm is the Euclidean distance, but, in a broader sense, it may be a metric or arbitrary distance function.

In other words, MDS attempts to find an embedding from the ***I*** objects into $\mathbf{R}^N$ such that distances are preserved. If the dimension *N* is chosen to be 2 or 3, we may plot the vectors x_i to obtain a visualization of the similarities between the ***I*** objects. Note that the vectors x_i are not unique: With the Euclidean distance, they may be arbitrarily translated, rotated, and reflected, since these transformations do not change the pair wise distances $\left\| x_i - x_j \right\|$.

PRESENT WORK

Example 1

Let us consider the following matrix as higher dimensional data.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 4 & 5 & 6 & 7 & 1 \\ 3 & 4 & 5 & 6 & 7 & 1 & 2 \\ 4 & 5 & 6 & 7 & 1 & 2 & 3 \\ 5 & 6 & 7 & 1 & 2 & 3 & 4 \\ 6 & 7 & 1 & 2 & 3 & 4 & 5 \\ 7 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

Then this matrix has been reduced to lower dimensional data using three dimensionality reduction techniques such as singular value decomposition, principal component analysis and multi dimensional scaling. Clustering is performed on the original data and as well as the three lower dimensional data obtained through dimensionality reduction techniques using K-means algorithm of MATLAB with varying number of clusters.

Mean squared error obtained from all the four datasets (original data set and three reduced data sets) is tabulated. From the tabulated values it is very clearly evident that clustering performed on the data obtained through Multidimensional scaling is same as the clustering performed on the original data but the order of the mean squared error values differ due to the random selection of cluster centers.

Table 1: Mean Squared Error Obtained by Performing Clustering on Original Data Set (7X7 Matrix) and the Three Data Sets Obtained by Applying SVD, PCA and MDS on Original Data Set

Number of Clusters	Mean Squared Error obtained from Original Data	Mean Squared Error obtained After Applying SVD on the Original Data	Mean Squared Error obtained After Applying PCA on the Original Data	Mean Squared Error obtained After Applying MDS on the Original data
K=2	21.0000 39.2000	0 116.9770	39.2000 21.0000	21.0000 39.2000
K=3	16.5000 15.0000 0	36.3993 0 0	5.3333 21.0000 10.0000	16.5000 0 15.0000
K=4	0 10.0000 0 5.3333	16.8285 0 0 0	0.0000 10.0000 5.3333 0.0000	0 5.3333 10.0000 0
K=5	5.3333 0 0 0 0	0 0 7.1283 0 0	0.0000 0.0000 0.0000 5.3333 0.0000	0.0000 0 5.3333 0.0000 0
K=6	0 0 0 1 0 0	0 0 0 0 2.0243 0	0.0000 1.0000 0.0000 0.0000 0.0000 0.0000	0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
K=7	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0

Example 2

Let us consider the following matrix as higher dimensional data.

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\ 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 & 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\ 81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 & 91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100 \\ 101 & 102 & 103 & 104 & 105 & 106 & 107 & 108 & 109 & 110 & 111 & 112 & 113 & 114 & 115 & 116 & 117 & 118 & 119 & 120 \\ 121 & 122 & 123 & 124 & 125 & 126 & 127 & 128 & 129 & 130 & 131 & 132 & 133 & 134 & 135 & 136 & 137 & 138 & 139 & 140 \\ 141 & 142 & 143 & 144 & 145 & 146 & 147 & 148 & 149 & 150 & 151 & 152 & 153 & 154 & 155 & 156 & 157 & 158 & 159 & 160 \\ 161 & 162 & 163 & 164 & 165 & 166 & 167 & 168 & 169 & 170 & 171 & 172 & 173 & 174 & 175 & 176 & 177 & 178 & 179 & 180 \\ 181 & 182 & 183 & 184 & 185 & 186 & 187 & 188 & 189 & 190 & 191 & 192 & 193 & 194 & 195 & 196 & 197 & 198 & 199 & 200 \end{pmatrix}$$

Then this matrix has been reduced to lower dimensional data using three dimensionality reduction techniques such as singular value decomposition, principal component analysis and multi dimensional scaling. Clustering is performed on the original data and as well as the three lower dimensional data obtained through dimensionality reduction techniques using K-means algorithm of MATLAB with varying number of clusters. Mean squared error obtained from all the four datasets (original data set and three reduced data sets) is tabulated. From the tabulated values it is very clearly evident that clustering performed on the data obtained through Multidimensional scaling is same as the clustering performed on the original data but the order of the mean squared error values differ due to the random selection of cluster centers.

Table 2: Mean Squared Error Obtained by Performing Clustering on Original Data Set (10 X 20 Matrix) and the Three Data Sets Obtained by Applying SVD, PCA and MDS on Original Data Set

Number of Clusters	Mean Squared Error obtained from original data	Mean Squared Error obtained after Applying SVD on the original data	Mean Squared Error obtained after Applying PCA on the original data	Mean Squared Error obtained after Applying MDS on the original data
K=2	8000 8000	1.0e+003*1.4530 1.0e+003*0.0000	1.0e+004*8.0000 1.0e+004*8.0000	1.0e+004*8.0000 1.0e+004*8.0000
K=3	16000 40000 16000	1.0e-024*0.0000 1.0e-024*0.0000 1.0e-024*0.1289	1.0e+004*1.6000 1.0e+004*1.6000 1.0e+004*4.0000	1.0e+004*1.6000 1.0e+004*1.6000 1.0e+004*4.0000
K=4	16000 4000 16000 4000	1.0e-025*0.0000 1.0e-025*0.0000 1.0e-025*0.1780 1.0e-025*0.0000	1.0e+004*0.4000 1.0e+004*1.6000 1.0e+004*0.4000 1.0e+004*1.6000	1.0e+004*0.4000 1.0e+004*1.6000 1.0e+004*0.4000 1.0e+004*1.6000
K=5	4000 16000 0 4000 4000	1.0e-026*0.0000 1.0e-026*0.0000 1.0e-026*0.2687 1.0e-026*0.0000 1.0e-026*0.0000	1.0e+004*0.4000 1.0e+004*0.4000 1.0e+004*1.6000 1.0e+004*0.4000 1.0e+004*0.0000	1.0e+003*4.0000 1.0e+003*4.0000 1.0e+003*4.0000 1.0e+003*4.0000 1.0e+003*4.0000
K=6	160000 0 4000 0 4000 0	1.0e-026*0.0000 1.0e-026*0.0000 1.0e-026*0.0000 1.0e-026*0.0000 1.0e-026*0.1056 1.0e-026*0.0000	1.0e+003*4.0000 1.0e+003*4.0000 1.0e+003*0.0000 1.0e+003*0.0000 1.0e+003*4.0000 1.0e+003*4.0000	1.0e+003*0.0000 1.0e+003*4.0000 1.0e+003*4.0000 1.0e+003*4.0000 1.0e+003*0.0000 1.0e+003*4.0000

Table 2: Contd.,				
K=7	4000	1.0e-027*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-027*0.0000	1.0e+003*4.0000	1.0e+003*4.0000
	0	1.0e-027*0.0000	1.0e+003*0.0000	1.0e+003*4.0000
	0	1.0e-027*0.3642	1.0e+003*4.0000	1.0e+004*0.0000
	0	1.0e-027*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-027*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	16000	1.0e-027*0.0000	1.0e+003*4.0000	1.0e+003*4.0000
K=8	4000	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*4.0000
	4000	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*4.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*4.0000
	0	1.0e-028*0.8144	1.0e+003*4.0000	1.0e+003*0.0000
K=9	0	1.0e-028*0.0000	1.0e+003*4.0000	1.0e+003*4.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	4000	1.0e-028*0.2644	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
	0	1.0e-028*0.0000	1.0e+003*0.0000	1.0e+003*0.0000
K=10	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

Example 3

An image of 1219 ×1600 is considered and then the intensity of each pixel value is retrieved and stored in a matrix .It is treated as a higher dimensional data. Then this matrix has been reduced to lower dimensional data using three dimensionality reduction techniques such as singular value decomposition, principal component analysis and multi dimensional scaling. Clustering is performed on the original data and as well as the three lower dimensional data obtained through dimensionality reduction techniques using K-means algorithm of MATLAB with varying number of clusters. Mean squared error obtained from all the four datasets (original data set and three reduced data sets) is tabulated. From the tabulated values it is very clearly evident that clustering performed on the data obtained through Multidimensional scaling is same as the clustering performed on the original data but the order of the mean squared error values differ due to the random selection of cluster centers.

Table 3: Mean Squared Error Obtained by Performing Clustering on Original Data Set(1219 X 1600 Matrix) and the Three Data Sets Obtained by Applying SVD, PCA and MDS on Original Data Set

Number of Clusters	Mean Squared Error obtained from Original Data	Mean Squared Error obtained After Applying SVD on the Original Data	Mean Squared Error obtained After Applying PCA on the Original Data	Mean Squared Error obtained After Applying MDS on the Original Data
K=2	1.0e+004*2.9334 1.0e+004*4.2951	1.0e+004*0.0000 1.0e+004*7.2426	1.0e+004*4.7754 1.0e+004*2.4559	1.0e+004*2.9334 1.0e+004*4.2951
K=3	1.0e+004*2.0894 1.0e+004*2.0815 1.0e+004*2.6050	1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*6.2582	1.0e+004*3.4524 1.0e+004*2.2991 1.0e+004*1.1500	1.0e+004*2.6050 1.0e+004*2.0894 1.0e+004*2.0815
K=4	1.0e+004*0.7671 1.0e+004*2.5085 1.0e+004*2.1082 1.0e+004*0.9453	1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*5.6752	1.0e+004*0.9321 1.0e+004*1.6527 1.0e+004*1.3525 1.0e+004*2.3844	1.0e+004*0.6748 1.0e+004*0.9326 1.0e+004*2.9102 1.0e+004*1.9805
K=5	1.0e+004*0.8280 1.0e+004*2.0766 1.0e+004*0.7025 1.0e+004*0.6823 1.0e+004*1.8124	1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*5.1817 1.0e+004*0.0000	1.0e+004*0.9172 1.0e+004*1.4213 1.0e+004*0.7095 1.0e+004*0.7555 1.0e+004*2.2606	1.0e+004*0.7733 1.0e+004*2.0417 1.0e+004*0.5149 1.0e+004*1.8520 1.0e+004*0.8943
K=6	1.0e+004*0.5546 1.0e+004*0.2453 1.0e+004*3.0308 1.0e+004*0.1309 1.0e+004*0.7409 1.0e+004*1.2877	1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*4.8349 1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*0.0000	1.0e+004*1.5044 1.0e+004*1.9478 1.0e+004*1.0078 1.0e+004*0.8599 1.0e+004*0.2967 1.0e+004*0.4466	1.0e+004*0.7189 1.0e+004*0.8681 1.0e+004*1.8308 1.0e+004*1.0212 1.0e+004*0.8550 1.0e+004*0.4647
K=7	1.0e+004*1.7249 1.0e+004*0.3662 1.0e+004*2.1066 1.0e+004*0.5507 1.0e+004*0.4683 1.0e+004*0.4930 1.0e+004*0.1134	1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*4.5322 1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*0.0000 1.0e+004*0.0000	1.0e+004*1.0303 1.0e+004*0.8240 1.0e+004*0.9516 1.0e+004*1.1395 1.0e+004*0.8325 1.0e+004*0.7601 1.0e+004*0.2237	1.0e+004*0.6181 1.0e+004*0.6296 1.0e+004*1.4063 1.0e+004*0.2865 1.0e+004*0.2226 1.0e+004*1.8589 1.0e+004*0.6581

The time taken to convert the above mentioned image of size 1219×1600 (higher dimensional data) in to lower dimensional data using Singular Value Decomposition, Principal Component Analysis and Multidimensional Scaling in MATLAB are given below.

Table 4: Time Taken for Converting the High Dimensional Data to Low Dimensional Data through Three Dimensionality Reduction Techniques such as SVD, PCA and MDS

Time Taken to Convert the Higher Dimensional Data in to Lower Dimensional Data Using Singular Value Decomposition in Seconds	Time Taken to Convert the Higher Dimensional Data in to Lower Dimensional Data Using Principal Component Analysis in Seconds	Time Taken to Convert the Higher Dimensional Data in to Lower Dimensional Data Using Multidimensional Scaling in Seconds
24.844000 seconds	24.157000 seconds	24.953000 seconds

The time taken to perform cluster analysis using K-means algorithm on the original data as well as the lower dimensional data obtained through SVD, PCA and MDS are given below.

Table 5: Time Taken to Perform Cluster Analysis on the Original Data as Well as the Lower Dimensional Data Obtained through SVD, PCA and MDS

Number of Clusters (k)	Time Taken to Perform Cluster Analysis on the Original Data in Seconds	Time Taken to Perform Cluster Analysis on the Lower Dimensional Data Obtained through SVD in Seconds	Time Taken to Perform Cluster Analysis on the Lower Dimensional Data Obtained Through PCA in Seconds	Time Taken to Perform Cluster Analysis Lower Dimensional on the Data Obtained through MDS in Seconds
K=2	1.484000	1.500000	9.625000	2.563000
K=3	4.079000	3.000000	28.390000	3.656000
K=4	3.735000	3.671000	25.547000	4.218000
K=5	4.437000	4.906000	34.328000	4.016000
K=6	4.797000	5.578000	23.39000	6.813000
K=7	5.953000	7.375000	23.406000	12.360000

The total time required for converting the higher dimensional data in to lower dimensional data and performing cluster analysis on the data is tabulated below.

Table 6: Time taken to Perform Data Reduction and as Well as Cluster Analysis

Number of Clusters (k)	Total time taken to perform cluster analysis on the original data in seconds	Total time taken to perform data reduction using SVD and to perform cluster analysis on the lower dimensional data obtained through SVD in seconds	Time taken to perform data reduction through PCA and to perform cluster analysis on the lower dimensional data obtained through PCA in seconds	Total time taken to perform data reduction through MDS and to perform cluster analysis on the lower dimensional data obtained through MDS in seconds
K=2	1.484000	24.844000+1.500000 =26.344	24.157000+9.625000 =33.782	24.953000+2.563000 =27.516
K=3	4.079000	24.844000+3.000000 =27.844	24.157000+28.390000 =52.547	24.953000+3.656000 =28.609
K=4	3.735000	24.844000+3.671000 =28.515	24.157000+25.547000 =49.704	24.953000+4.218000 =29.171
K=5	4.437000	24.844000+4.906000 =29.75	24.157000+34.328000 =58.485	24.953000+4.016000 =28.969
K=6	4.797000	24.844000+5.578000 =30.422	24.157000+23.39000 =47.547	24.953000+6.813000 =31.766
K=7	5.953000	24.844000+7.375000 =32.219	24.157000+23.406000 =47.563	24.953000+12.360000 =37.313

From the above table it is clearly evident that Singular Value Decomposition takes less time than the other two dimensionality reduction techniques (PCA and MDS) to perform cluster analysis.

CONCLUSIONS

It is very difficult and time consuming to perform clustering on a higher dimensional data due to large number of dimensions. If the higher dimensional data can be reduced to lower dimensional data, clustering can be performed easily.

In order to convert the higher dimensional data in to lower dimensional data, a number of dimensionality reduction techniques have been developed. In this work we have been interested in three dimensionality reduction techniques and they are singular value decomposition, principal component analysis and multidimensional scaling. Different types of higher dimensional data have been considered and it has been reduced to lower dimensional data through the three above mentioned dimensionality reduction techniques. Cluster analysis has been done on the original data as well as the three lower dimensional data obtained through reduction techniques. It is very clear from the tabulated values that cluster analysis performed on the original data is almost same as the cluster analysis performed on the data obtained through multidimensional scaling. It is also evident from the tabulated values that the time taken to perform cluster analysis on the data set reduced through singular value decomposition takes less time than the time taken to perform cluster analysis on the data sets reduced through principal component analysis and multidimensional scaling. This work can be extended to other dimensionality reduction techniques.

REFERENCES

1. *EkezieDanDan : "Principal component analysis, an aid to interpretation of data. A case study of oil palm". Journal of Emerging trends in Engineering and Applied Sciences (JETEAS) 4(1), 73-76*
2. *T.D.Venkateswaran, G.Arumugam : "Defect detection in fabric images using singular value decomposition technique". International Journal on Computer Technology and Applications, Vol-5(2), 351-356*
3. *Orly Alter, Patrick O.Brown and David Botstein (August 29, 2000) : "Singular value decomposition for genome-wide expression data processing and modeling". PNAS, Vol.97, No.18, 10101-10106*
4. *John Mandel (February 1982) : "Use of the singular value decomposition in regression analysis". The American Statistician, Vol- 36, No.1*
5. *Massimo Franceschet (2009) : "A Cluster analysis of scholar and journal bibliometric indicators". Journal of the American Society for Information Science and Technology, 60(10)*
6. *Huizhou, Trevor Hastie and Robert Tibshirani : "Sparse principal component analysis". Journal of Computational and Graphical Statistics, Volume 15, Number 2, Pages 265-286*
7. *Tamilselvi Madeswaran, G.M.Kadhar Nawaz (December 2012) : "A Comparative analysis of classification of micro array gene expression data using dimensionality reduction techniques". International Journal of Computer and Electronics Research, Volume - 1, Issue – 4*
8. *Geoffrey J.Goodhill, Martin W.Simmen and David J.Willshaw : "An evaluation of the use of multidimensional scaling for understanding brain connectivity". Biological Sciences,Vol-348,No-1325,PP.265-280*